LISTing Newsletter

Newsletter of the
Long Island Sinclair\TimeH
Users Group

Incorporating NYTSE

October 1988

Next Meeting

November 13 2:00PM

Harvey Rait's House

L.I.S.T.
5 Peri Lane
Valley Stream, NY 11581

TO:

Don Lambert
3310 Clover Dr. S
Cedar Rapids, IA
52404

FIRST CLASS MAIL
DATE MEETING NOTICE
Please DON'T delay!

LISTING

Please send submissions to:

LIST
Mr. Harvey Rait
5 Peri Lane
Valley Stream, N.Y. 11581

COMING EVENTS:

November 13 1988 LIST Meeting.

MEETING MINUTES
October 15, 1988

This SWAPMEET though not as big as our last, offered many great buyes. Bargains such as a 2068 for $30.00, 2068 software $2.00 and ByteBack Modem for $15.00 were offered.

Harvey started the meeting following the SWAPMEET.
N.Y.T.S.E. meeting was held on Oct. 24, 7:00PM at Kim's Restaurant, Park Avenue South between 22 & 23rd St.
LIST members and Sinclair users are invited to attend. (Future NYTSE meeting dates will be published in advance.-EDITOR)

Harvey announced #10LIST Tape should be coming soon, and a QL Program tape is forthcoming.

LIST member Mike Fink is in the hospital with a gall bladder condition, we all wish him a speedy recovery.
Mike has equipment for sale, and is also seeking some.
See his ad in our new Classified Section.

Stoney informed us that A+ and Curry have sold out all QL's. Curry advised that chips needed for QL repair can no longer be obtained. Specifically #8301, & #8302. Curry also recommends people do not upgrade or add to their QL system do to the chip shortage problem.

Brooklyn Closeout Corp. may still have TS2068, 1500, & 1000 for sale.

SPOTLIGHT ON MEMBERS

Jay Siegal - 1000 & 2068; Jay has developed software for the 2068 such as DBASE 2
John Falto - 1000, 1500 & 2068 John uses his computer for business and has developed his own software.
Greg Papovich - Sysop WYSIWYG (What You See Is What You Get) 201-956-7703, 24Hour300, 1200, & 2400 Baud.
Edgar Gross - 1500 &2068 Edgar has adapted his 2068 to a RGB monitor.

CLASSIFIEDS

This Classified Section is available to all LISTING subscribers, FREE ofcharge. The only restriction is that it is to be used only for the seeking, selling or swapping of Sinclair, Timex or Microace computer equipment, periferals and software.

Mike Fink is selling a brandnew GEMINI 10X printer, and is seeking a COBOL-COMPILER for the 2068.
Fred Stern is seeking the program required to use the 2050 Modem, with theTS1000.
He is also seeking aSuntronics, or other keyboard for the 1000.

Edgar Gross is seeking a disc system for the 2068.

To place or answer a classified

LISTING
FRED STERN
214 ROBERTS ST.
HOLBROOK, N.Y. 11741

A Final Word

My name is Fred Stern, and I am the editor of this edition of LISTING. I would like to thank Mr. Tom Skapinski for his helping me with my printer driver, for his assistance in proofing the newsletter, and for his time and trouble to get the newsletter printed, corallated, and mailed. I also thank Mr. Steve Kaye who is your recording secretary / reporter for this edition of LISTING, and extend condolences to Steve on the recent passing of his mother.
Finally, I would like to thank Mr. George Gilder for sending in the two enclosed articles.

EXTRA EXTRA EXTRA EXTRA EXTRA
Tom Skapinski has advised me that SCLD chips used in the TS2068 may be obtained from the following sources;
Eric Johnson
249 N. Harden Avenue
Orange City, Fl 32763
904-775-4935

Paul Holmgren
5231 Wilton Wood Ct.
Indianapolis, In 46254
317-291-6002
Write or call for information and prices.
Dan Elliot of Promise Land Electronics is negotiating with Timex of Portugal for a bulk buy of SCLD chips. If he is successfull he intends to offer for sale about half of the chips.
ZX Appeal reported the TS1500 with TS2040 can be purchased for $30.00 from;
Electronics Surplus Inc.
1224 Prospect Avenue
Cleveland, Ohio
216-621-1052
This sounds too good to be true.

CUSTOM COLORS FOR THE 2068

****

By using a combination of PAPER, INK and UDG screens we can have an almost endless variety of colors.

Most color printing we see in magazines, packages etc. are printed using only four colors; cyan,yellow,magenta and black. These are called process colors.

The seemingly endless varieties are determined by a combination of screened and solid colors.The screens are percentages of the 4 process colors. The following programs use similar but simplified features.

The first listing (C1) is from a Res-istor Color Code program I have developed. The screen dump shows the custom colors. Below are the UDG's developed for this and the following program.

Note that lines 407 to 411 use the combinations I noted above.

Color Blocks uses the same UDG's as the prior program. An endless chain of colors, hues and their shades can be seen. By changing or adding your own screens the range increases greatly.

I can answer questions with SAE.
George Gilder 67-38 108 Street,
Forest Hills,NY 11375

```
RESISTOR COLOR CODE CHART
Enter color bars on resistor

▓ BLACK=0
▒ BROWN=N
▓ RED   =R
▒ ORANG=O
▓ YELLW=Y
▓ GREEN=G
▓ BLUE  =B
▒ VIOLT=V
▒ GRAY  =E
□ WHITE=W
```

```
▒ GOLD   =L
▒ SILVER=S

UDG ▓ ≡ ▒ □ |  | ▓
    O P Q R S T U
-------------------------------
SCREEN DUMP
```

```
404 LET X=0: GO SUB 462: REM
     *******************
     *    COLORS  C1    *
     * ©1987 G. GILDER  *
     ********************
405 CLS : BORDER VAL "7": PAPER
VAL "7": INK X: LET F=1
406 PRINT INK 2;"RESISTOR COLOR
CODE CHART"; INK 0';"Enter colo
r bars on resistor"''
407 PRINT INK X;TAB X;"▓ BLACK=
0";TAB X; INK 2; PAPER 0;"▒"; PA
PER 7; INK 2;" BROWN=N"; INK 2;T
AB X;"▓ RED   =R";
408 PRINT TAB X; PAPER 2; INK 6
;"▒"; PAPER 7; INK 2;" ORANG=O"
409 PRINT TAB X; INK 6;"▓ YELLW
=Y";TAB X; INK 4;"▓ GREEN=G";TAB
 X; INK 1;"▓ BLUE  =B"; 
410 PRINT TAB X; PAPER 3; INK 1
;"▒"; PAPER 7;" VIOLT=V";TAB X;
INK X;"▒ GRAY  =E";TAB X;"□ WHITE
=W";
411 PRINT ''''TAB X;; PAPER 6;
INK 0;"▒"; PAPER 7; INK 6;" GOLD
  =L";TAB X; INK X;"▒"; INK 0;"
SILVER=S";
412 GO SUB 457
413 PRINT AT 20,0;"UDG ▓ ≡ ▒ □
|  | ▓ "'"      O P Q R S T U": STO
P
457 PRINT AT VAL "14",VAL "12";
"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
458 PLOT VAL "96",VAL "64": DRA
W VAL "80",X: DRAW X,VAL "-9": D
RAW VAL "-80",X: DRAW X,VAL "9"
460 PLOT 96,60: DRAW -30,0: PLO
T 176,60: DRAW 30,0
461 RETURN
462 RESTORE : FOR F=X TO VAL "3
9": READ Y: POKE USR "O"+F,Y: NE
XT F
463 DATA INT 85,170,85,170,85,1
70,85,170,255,129,129,129,129,12
9,129,255,128,128,128,128,128,12
8,128,128,1,1,1,1,1,1,1,1,170,17
0,170,170,170,170,170,170,170
464 LET h=255: LET g=170: RESTO
RE 465: FOR F=X TO VAL "15": REA
D Y: POKE USR "O"+F,Y: NEXT F
465 DATA g,h,g,h,g,h,g,h,255,0,
255,0,255,0,255,0
466 RETURN
490 REM


500 REM ** COLOR BLOCKS**
505 LET a=5: FOR p=157 TO 160:
FOR i=0 TO 6
510 LET k=INT (RND*7): PRINT AT
 a,i*2+6; PAPER k; INK i;CHR$ p;
CHR$ p
515 NEXT i: LET a=a+1: NEXT p
550 PRINT AT 20,6;"PRESS ANY KE
Y"';TAB 7;"TO CONTINUE": PAUSE 0
: GO TO 500
590 REM


600 REM ********************
        * Custom Colors C2  *
        *********************
605 REM Try this program & see
     dozens of color bars using
     only one screen and the
     standard 2068 colors. NOTE
     Type and run these
     programs together.
610 LET b$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒"
620 FOR i=0 TO 7: FOR f=0 TO 6:
PRINT PAPER f; INK i;b$
630 NEXT f: NEXT i
```

```
 500 IF A=1 THEN PRINT AT H,G; I
NK 2; PAPER X;"▒▒"
 510 IF A=2 THEN PRINT INK 2;AT
H,G;"▒▒"
 520 IF A=3 THEN PRINT AT H,G; P
APER 2; INK 6;"▒▒"
 530 IF A=4 THEN PRINT AT H,G; I
NK 6;"▒▒"
 540 IF A=5 THEN PRINT AT H,G; I
NK 4;"▒▒"
 550 IF A=6 THEN PRINT AT H,G; I
NK 1;"▒▒"
 560 IF A=7 THEN PRINT AT H,G; P
APER 3; INK 1;"▒▒"
 570 IF A=8 THEN PRINT AT H,G;"▒
▒"
 580 IF A=13 AND F=3 THEN PRINT
AT H,G; INK X;"▒▒▒"
 590 IF A=9 THEN PRINT AT H,G;"|
|"
 600 IF A=12 AND F=3 THEN PRINT
AT H,G; PAPER 6; INK 0;"▒▒"
 610 RETURN
 620 PRINT
 630 LET A$=STR$ A: LET C$=""
 640 GO SUB 730
 650 RETURN
 660 PRINT AT VAL "14",VAL "12";
"▒▒▒▒▒▒▒▒▒▒▒▒"
 670 PLOT VAL "96",VAL "64": DRA
W VAL "80",0: DRAW 0,VAL "-9": D
RAW VAL "-80",0: DRAW 0,VAL "9"
 680 PLOT 96,60: DRAW -30,0: PLO
T 176,60: DRAW 30,0
 690 RETURN
 700 RESTORE : FOR F=0 TO VAL "3
9": READ Y: POKE USR "0"+F,Y: NE
XT F
 710 DATA INT 85,170,85,170,85,1
70,85,170,255,129,129,129,129,12
9,129,255,128,128,128,128,128,12
8,128,128,1,1,1,1,1,1,1,1,170,17
0,170,170,170,170,170,170,170,170
 720 RETURN
 730 IF A$="1" THEN LET A$="0"
 740 IF A$="11" THEN LET A$=".0"
 750 IF A$="2" THEN LET A$="00"
 760 IF A$="3" THEN LET A$="000"
 770 IF A$="4" THEN LET A$="0000
"
 780 IF A$="5" THEN LET A$="0000
0"
 790 IF A$="6" THEN LET A$="0000
00"
 800 IF A$="7" OR A$="8" OR A$="
9" THEN LET A$="": PRINT AT H,G;
"▒▒": GO TO 220
 810 IF A$="13" THEN LET A$="/10
0"
 820 IF A$="12" THEN LET A$="/10
"
 830 LET B$(3 TO )=A$: RETURN
 840 GO SUB 970
 850 IF A=11 THEN LET A=0
 860 IF A=12 THEN LET A=-1
 870 IF A=13 THEN GO TO 960
 880 LET A=A+4
 890 DIM S(24): FOR F=1 TO 48
 900 IF F>24 THEN GO TO 960
 910 LET S(F)=R(F)*M(A)
 930 IF VAL Q$>47E5 THEN GO TO 9
60
 940 IF VAL Q$=S(F) THEN RETURN
 950 NEXT F
 960 PRINT INK VAL "1"; FLASH VA
L "1";"NOT STANDARD 5/10 % VALUE
": RETURN
 970 IF B$(1)="0" THEN LET B$=B$
(2 TO )
 980 LET q$="": FOR f=1 TO LEN b
$
 990 IF CODE b$(1)<>0 AND (f)=46
 THEN LET Q$=Q$( TO 2): RETURN
1000 IF CODE b$(f)=32 THEN GO TO
 1020
1010 LET q$=q$+b$(f): NEXT F: RE
TURN
1020 IF Q$(3)="." THEN LET Q$=Q$
( TO 2)
1030 IF Q$(3 TO )="/10" THEN LET
 Q$=Q$(1)+"."+Q$(2)
1040 IF Q$(3 TO )="/100" THEN LE
T Q$="."+Q$( TO 2)
1050 IF B$(3 TO 6)="/100" THEN P
RINT AT 16,12;".";AT 16,20;"
": RETURN
1060 IF B$(3 TO 5)="/10" THEN PR
INT AT 16,16;".";AT 16,20;"    "
1070 RETURN
1100 REM Standard 5% E-24 values
1110 DATA 1.0,1.1,1.2,1.3,1.5,1.
6,1.8,2.0,2.2,2.4
1120 DATA 2.7,3.0,3.3,3.6,3.9,4.
3,4.7,5.1,5.6,6.2
1130 DATA 6.8,7.5,8.2,9.1
9000 CLEAR : SAVE *"RG" LINE 1:
CAT.*
```

RESISTOR GUIDE

I am sure you have seen other programs that read resistor values. This one is different in several respects. It shows the resistor in exact colors. It runs the results through a "diction ary" of values & tells you if the value is legal.

Press the color bands, one at a time. The computer recognizeS Silver & gold bands.

Screen dump 1 shows a resistor that has passed. No. 2 is a resistor that is not a standard value.

I have developed faster resistor programs using similar techniques, using MC,etc. I chose this because it's easier to type in.

All inquiries welcome with SSAE please. George Gilder,67-38 108 th St. #D 61, Forest Hills, NY.

```
RESISTOR COLOR CODE CHART
Enter color bars on resistor

 BLACK=0
 BROWN=N
 RED  =R
 ORANG=O
 YELLW=Y
 GREEN=G
 BLUE =B
 VIOLT=V
 GRAY =E
 WHITE=W

          2  2  00

 GOLD  =L
 SILVER=S

ANOTHER RESISTOR ? (Y N)
```

RESISTOR SCREEN DUMP 1

```
  10 REM ************************
  20 REM *RESISTOR COLOR CODES*
  30 REM * WITH E-4 5% CHECKER*
  40 REM *  ©1987 G. GILDER   *
         ************************
  50 REM DIM r(96): DIM m(7)
  60 DIM r(24): DIM m(9)
  70 RESTORE 1090: FOR x=1 TO 24
  80 READ r(x)
  90 NEXT x
 100 LET m(1)=.01: LET m(2)=.1:
LET m(3)=1: LET m(4)=10: LET m(5
)=100
 110 LET m(6)=1000: LET m(7)=100
00: LET M(8)=100000: LET M(9)=10
00000
 120 POKE 23658,8
 130 GO SUB 700
 140 LET X=0: DIM B$(10): CLS :
BORDER VAL "7": PAPER VAL "7": I
NK X: LET F=1
 150 PRINT INK 2;"RESISTOR COLOR
 CODE CHART"';"Enter color bars
on resistor"''
 160 PRINT INK X;"▓ BLACK=0"' IN
K 2; PAPER X;"▒"; PAPER 7; INK 2
;" BROWN=N"; INK 2'"▓ RED  =R"
 170 PRINT PAPER 2; INK 6;"▒"; P
APER 7; INK 2;" ORANG=O"
 180 PRINT INK 6;"▓ YELLW=Y"' IN
K 4;"▓ GREEN=G"' INK 1;"▓ BLUE =
B"
 190 PRINT PAPER 3; INK 1;"▒"; P
APER 7;" VIOLT=V"' INK X;"▒ GRAY
 =E"'"□ WHITE=W"
 200 PRINT '''' PAPER 6; INK X;"
▒"; PAPER 7; INK 6;" GOLD  =L"'
INK X;"▒"; INK X;" SILVER=S"
 210 GO SUB 660
 220 LET A=0: LET a$=INKEY$: IF
a$="" THEN GO TO 220
 230 IF A$="0" THEN LET A=10
 240 IF a$="N" THEN LET a=1
 250 IF a$="R" THEN LET a=2
 260 IF a$="O" THEN LET a=3
 270 IF a$="Y" THEN LET a=4
 280 IF a$="G" THEN LET a=5
 258 IF a$="B" THEN LET a=6
 268 IF a$="V" THEN LET a=7
 278 IF a$="E" THEN LET a=8
 320 IF a$="W" THEN LET a=9
 330 IF a$="L" AND F=3 THEN LET
a=12
 340 IF a$="S" AND F=3 THEN LET
a=13
 350 IF NOT A THEN GO TO 220
 360 IF F=1 THEN PRINT AT 14,X;
 370 GO SUB 480
 380 IF F<>3 AND a=10 THEN LET a
=0
 390 LET B$(F)=STR$ A
 400 IF f=3 AND a=10 THEN LET a=
11
 410 IF F=3 THEN GO SUB 630: PRI
NT AT 16,F*3+11;B$(F TO ): GO SU
B 840: GO TO 440
 420 PRINT AT 16,F*3+11;B$(F): L
ET F=F+1
 430 LET A$="": PAUSE 30: GO TO
220
 440 PRINT AT 21,0;"ANOTHER RESI
STOR ? (Y N)"
 450 LET 0$=INKEY$: IF 0$="" THE
N GO TO 450
 460 IF 0$="N" THEN STOP
 470 GO TO 140
 480 LET H=14: LET G=F*3+10
 490 IF A=10 THEN PRINT AT H,G;"
▓▓"
```

ZX-81 PARALLEL PRINTER INTERFACE

ONE OF THE MOST PRACTICAL ACCESSORIES FOR ANY COMPUTER is a conventional printer, but some computers were not initially designed to work with a standard "Centronics-type" parallel printer—which is the kind that's usually available at low to moderate cost.

The Sinclair ZX-81 was among those personal computers that made no provision for interfacing with conventional printers. But if you're into a ZX-81, and have looked longingly at those $100 standard-paper printers, here's your chance to upgrade your computer with a standard printer interface.

Centronics Compatible

The interface shown in the photographs—and its driver program—allows the ZX-81 to operate any printer that is "Centronics-compatible." In keeping with the Sinclair philosophy of making it small, simple and inexpensive, the interface uses five commonly available integrated circuits and a few capacitors. The most expensive parts are the connectors. If you have a well-stocked junkbox, or the extreme patience to solder ribbon cables, the project will cost you under $20.

How It Works

To understand a parallel interface, let's look at what the printer expects and what it does in response. The important parallel printer connections are eight (sometimes seven) data lines that input character codes, a STROBE connection, and the BUSY and/or the ACK *handshake*. Other connections might be provided for control purposes, but those are absolutely required.)

The computer tells the printer that new data is available through a short negative pulse on the STROBE connection. The printer reacts by bringing the BUSY connection high to stop output from the computer, and then processes the character (it may print it immediately or store it in an internal buffer memory and print later.) When the printer is finished processing, it produces an ACK(nowledge) pulse similar to the STROBE and brings the BUSY low. Obviously, ACK and BUSY convey similar information so only one need be monitored. (Most personal computers recognize the BUSY rather than the ACK.)

The ZX-81 printer interface shown in Fig. 1 has this general design: It uses 8 data lines for the character code (Centronics connector pins 2-9), produces a STROBE pulse (pin 1), and monitors BUSY (pin 11) to determine when to send the next character. ACK (pin 10) isn't used. Those connector pins are "standard" and should be compatible with virtually all parallel printers; but just to be safe, check your printer's manual for unusual pinouts.

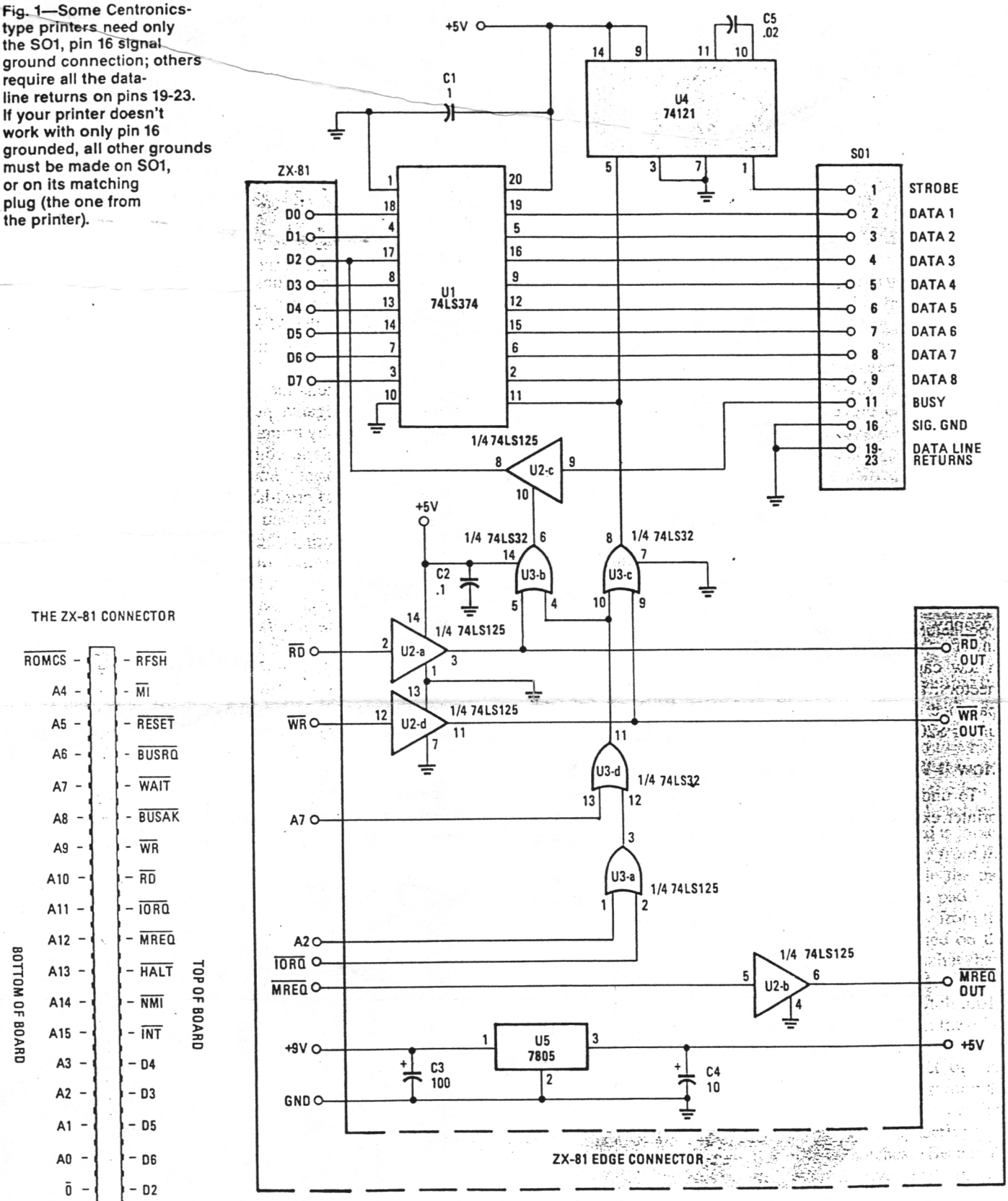
The output port is a 74LS374 octal dual flip-flop which will latch the data from the computer on the printer data lines

By Rudy Massey

Now you can use any standard parallel printer with your ZX-81 home computer.

NOVEMBER 1986

Fig. 1—Some Centronics-type printers need only the SO1, pin 16 signal ground connection; others require all the data-line returns on pins 19-23. If your printer doesn't work with only pin 16 grounded, all other grounds must be made on SO1, or on its matching plug (the one from the printer).



THE ZX-81 CONNECTOR

Fig. 2—The ZX-81's connections on the top and bottom of the system connector are different. Make certain that you get them right the first time—you might not get a second chance.

PARTS LIST FOR
ZX-81 PARALLEL PRINTER INTERFACE

SEMICONDUCTORS

U1—74LS374 octal dual flip-flop integrated circuit
U2—74LS125 three-state quad buffer integrated circuit
U3—74LS32 quad 2-in OR gate integrated circuit
U4—74121 monostable multivibrator integrated circuit
U5—7805 5-volt regulator

CAPACITORS

C1, C2—0.1-μF, 50-WVDC disc capacitor
C3—100-μF, 16-WVDC electrolytic capacitor
C4—10-μF, 16-WVDC tantalum capacitor
C5— 02μF disc capacitor

ADDITIONAL PARTS AND MATERIALS

SO1—Centronics-type connector, solder type
Enclosure/PC Board (Radio Shack 270-291)
50 pin card edge connector
12-inches 50-conductor ribbon cable
6-10 feet 20-conductor ribbon cable
46 pin section of 0.1" spaced double row header
1½" wide section of double-sided 0.1" spaced card edge
Wire-wrap materials, bus wire, solder

OPTIONAL PARTS AND MATERIALS

20-pin header connector 50 pin header connector
20-pin double-row right angle wire-wrap header
50-pin double-row wire-wrap header
The following are available from Zebra Systems, 78-06 Jamaica Ave., Woodhaven, NY 11421.
C110 Edge connector—$5.95
C111 Protocard—$9.95
C112 Extension connection—$2.50
Please add $3 per total order for shipping and handling (only via UPS). Additional $3 for COD via UPS.

when a falling pulse is sensed on its control line. A STROBE pulse is produced by a 74121 monostable multivibrator set up to trigger on a rising edge (note only one capacitor is needed with this IC to set the duration). By using the same negative going pulse to trigger both, an output to this port signals the printer automatically. The BUSY line is monitored by a one bit input port formed by a 74lS125 quad buffer. This IC has three-state outputs that isolate the gate from data line D2 when the control line is high. To define these ports as in/out, RD and WR are OR'ed to the enable lines (after buffering to reduce loading due to the piggy-backed memory) via a 74LS32 quad OR gate. One of the remaining gates must be used for IORQ (to differentiate from memory operations) and the other two are for address-line decoding. Both input and output can use the same address.

Port Usage

Here we must pause to mention the port usage of the ZX-81. Although the Z-80A microprocessor inside can address any of 256 input/output ports defined by decoding lines A0 through A7, eight ports can be had with no decoding: just activate one device off each line. Sinclair completely decoded only one address, FF(hex) or all lines high. That, plus FE(hex) and FD(hex), which correspond to A0 = low and A1 = low respectively, are used internally. FB(hex) is used for the output to the Timex/Sinclair printer, decoded by the printer as A2 = low and A7 = high. That leaves available A2 = low and A7 = low, which is appealing in our case because there are just enough inputs to decode A2 and A7 low. Thus, 7B(hex) will avoid all "taboo" ports and still leave plenty available for other devices.

It will be apparent if you look closely that it doesn't matter which input A2, A7, or IORQ use since they all must be low. But if A2 and A7 occupied the same gate, it would change states much more often than necessary. Using A2 and IOREQ together eliminates that. It might be nit-picking, but if the choice is ours—why not? MREQ was buffered with the one remaining gate but any choice is fine.

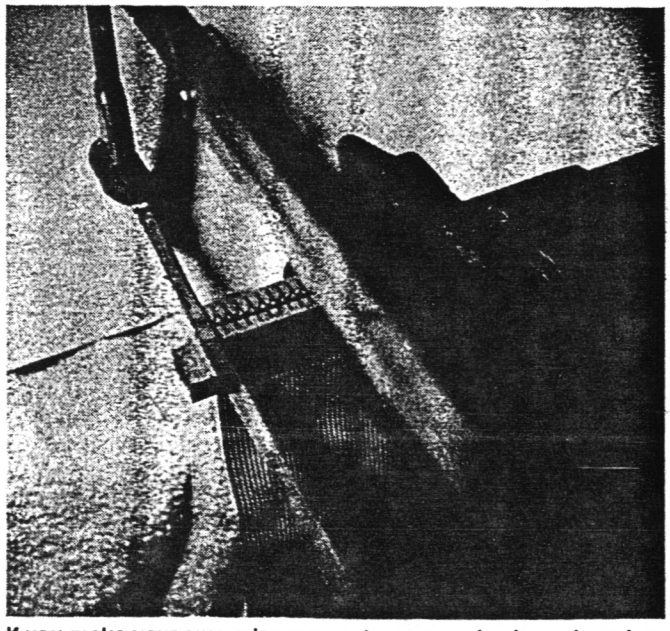
That's all there is to the circuit! Outputting a character code to port 7B(hex) will send it to the printer. Input on the same address; and if bit 2 is high, the printer is busy.

Construction

Because the ZX-81 is almost certain to have a memory pack attached, and Sinclair's own pack—the one often supplied with the computer—has no output to allow piggybacking peripherals, it seemed necessary to provide one in the interface's design. That probably doubles the complexity of construction; so if you have no need of an output, do yourself a favor and eliminate it (and all those connections). Likewise there is a 5-VDC line available on the ZX-81's output which will power the device (the interface is only 50 ma. or so), but tests showed a great increase in TV interference when it was used. If TVI is of no concern, you should be able to eliminate the regulator circuit (keep the 10-μF Tantalum capacitor on the 5-volt line).

Initially, a small perfboard/box combination from Radio Shack seemed perfect: All hardware is included and the board's holes are copper ringed, so point-to-point or wire-wrapped connections are possible. But solder the connections when strength or reliability is important.

(The space gets tight when the output connector is installed, so if you have trouble working in close places you might want to use a "protocard" and connector set specifically intended for homebrew ZX-81 projects. The parts are available from Zebra Systems, 78-06 Jamaica Ave., Woodhaven, NY 11421. See the parts list for additional details.)



If you make your own edge connectors, use a hacksaw to cut a standard connector down to size.

Let's start with the input connector. As shown in Fig. 2, the ZX-81 requires a non-standard card-edge connector that can be made from a 50 pin type. Before you clamp it on the ribbon cable, use needle-nose pliers to push out the two metal displacement pins on each end, then the third set in from what is to be the right side. (That is where the "key" will be.) Then use a small vise to evenly clamp the connector on a short piece of 50-wire ribbon cable.

Make a deep cut with a hacksaw on each end through the area where the pins were removed, then cut in from the side to provide clearance for the computer's cabinet. A small file can enlarge the space for the "key"—the key being a piece of plastic or circuit board that fits into the slot on the ZX-81's connector. The other end of the cable is soldered to the interface board, but a wire-wrap header would be easier (if you don't mind spending a few extra dollars). Keep in mind that the two wires on each edge aren't connected.

The output connection for the printer uses a 20 pin header, but direct soldering can be used. Since the BUSY line is needed—which is pin 11 on the Centronics connector—a 20 wire cable can't be used with an insulation displacement connector because only the first 10 pins would be used—the connection on pin 11 would be missed. A solder-type Centronics connector allows skipping pin 10 and using pin 11. (That's just one of many solutions. Use what you have so long as the necessary connections are made.)

Install the wire-wrap sockets with solder or glue in the positions shown (not much deviation is possible.) The output board can be cut from an old printed-circuit board or etched from scratch. Sandwich it between a section of dual header pins (it just fits) and solder. After trimming for proper fit on a connector or *rampack*, solder it to the board (there's a lot of stress here, so do a good job.) Next, install U5, the voltage regulator.

TABLE 1—LOADER PROGRAM

```
 10 REM ....(200 CHRS).....
.....
 20 LET A=16514
 30 SCROLL
 40 FOR B=1 TO 8
 50 INPUT A$
 60 IF A$="" THEN STOP
 70 IF LEN A$<>2 THEN GOTO 50
 80 POKE A,16*CODE A$(1)+CODE A
$(2)-476
 90 PRINT A$+" ";
100 LET A=A+1
110 NEXT B
120 IF A>=16699 THEN STOP
130 GOTO 30
140 REM RUN THIS FOR PRINTER
      TEST
150 LET A=16514
160 LET A$=STR$ A+" "
170 FOR Z=1 TO 8
180 LET B=PEEK A
190 LET C=INT (B/16)
200 LET D=B-(C*16)
210 LET A$=A$+CHR$ (C+28)+CHR$
(D+28)+" "
220 LET A=A+1
230 NEXT Z
240 IF A>16713 THEN STOP
250 LET A$=A$+CHR$ 8
260 RAND USR 16529
270 GOTO 160
```

TABLE 2—HEX CODE TO ENTER

21	D1	40	7E	EE	20	77	21
DF	40	7E	EE	20	77	C9	2A
10	40	EB	2A	14	40	E5	A7
ED	52	44	4D	E1	2B	2B	3E
46	ED	B9	E0	23	23	7E	A7
C8	47	23	23	7E	E5	CD	B7
40	E1	10	F7	C9	FE	80	20
04	D6	60	18	2E	A7	FE	1C
38	20	FE	26	30	04	C6	14
18	21	FE	40	30	04	C6	1B
18	19	FE	9C	D8	FE	A6	38
07	FE	C0	D0	D6	45	18	0B
D6	80	21	16	41	5F	AF	57
ED	5A	7E	FE	0D	20	05	CD
F6	40	3E	0A	F5	DB	7B	E6
04	20	FA	F1	D3	7B	C9	06
03	21	12	41	7E	A7	C8	CD
F6	40	AF	77	23	10	F5	C9
00	00	00	00	20	09	0E	14
0F	12	1B	08	0D	0A	0C	22
23	24	3A	3F	28	29	3E	3C
3D	2B	2D	2A	2F	3B	2C	40
27	5F	26	21	22	7C	25	5B
5D	25	00	00	00	00	00	00

Begin making the connections with the ground and power wires. Those lines are direct-soldered where possible, using bus wire. Next, connect all the capacitors (check polarity). At this point, it's a good idea to use a bench power supply to locate any problems while they can be easily fixed. Input about 8-VDC to the regulator and check for proper voltage at the sockets.

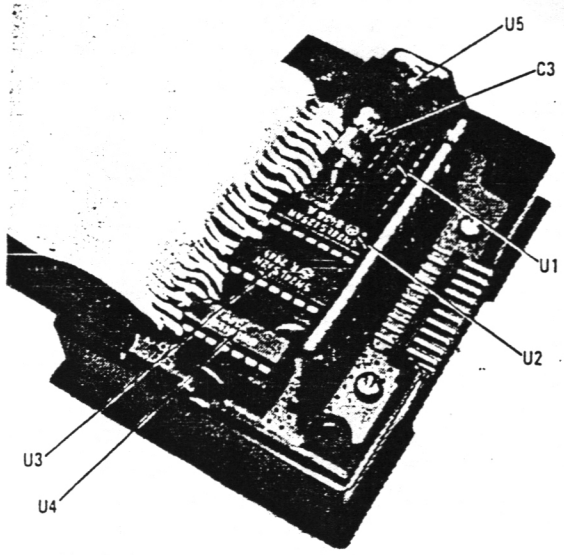
The wire-wrap pins are too long to fit inside the case so they must be trimmed to less than half their normal length. That doesn't leave much for multiple connections, so take extra care. Make the connections to the integrated circuit sockets first, using the schematic to identify the pins. Next, wire the output board to the input, and finally, the printer cable connector. Take your time; use a continuity tester to TRIPLE check your work, and *never* take any connection for granted. The ZX-81 is relatively tolerant of shorts but its output bus isn't buffered and major damage can occur. When it's all wired and you are sure it's correct, insert the integrated-circuits. Then connect the remaining components and power up the computer.

If you don't get a cursor on the screen in a reasonable amount of time, turn the power off and check your work. If there is a cursor, congratulations! Now you can fit the case.

The main cable enters through a small slot easily cut in the bottom case half. Likewise, the printer cable slot is totally in the bottom half.

The output board requires a hole in the top of the cabinet. (The plastic cabinet is soft and easily cut and filed.) Drill a few holes in the bottom of the cabinet for ventilation. If you have been careful, the interface will resemble a commercial product.

It's a tight fit, but the interface will fit a into a combination perfboard/box sold by Radio Shack. But you'll have to cut slots in the case for the connectors.



The Software

The program (listing in Table 1) searches the variable storage area for "A$" (any string can be substituted, more on that later) and outputs the whole string (256 bytes maximum) to the printer port. The graphics characters (Sinclair character codes 1-10) and the inverse numbers can be defined any way you want: as control codes, special characters, or whatever the printer will accept. This allows you to take advantage of the ZX-81's excellent string handling to do much of the work arranging and imbedding characters. Just define the string (no arrays, DIM A$ is out!), and use "RAND USR 16529" to print it.

To create the program, copy the program listing in Table 1 and use it to load the actual code given in Table 2. You will be left with a first line that looks like gibberish but actually contains the machine code. Now all lines except the first can be deleted. That *must* appear as the first line of any program using the interface.

A couple of utilities, that are also included for convenience, should be explained. As loaded, the program prints all "normal" ZX-81 characters as upper case, and all inverse characters as lower case that seems to work out best overall, but some programs need just the reverse. That is easily done by RAND USR 16514. The program is a toggle: Each time it is called it reverses the cases. Also included is a "just in case" program that outputs up to three bytes—starting at address 16658—directly to the printer. (It could be convenient if you don't want to redefine any characters for one-time use.) RAND USR 16641 calls it after you have POKED the codes in the addresses; those addresses are cleared after each use.

Address 16546 holds the code of the string printed; "A$" is 70(decimal), "B$" would be 71, "C$" 72, and so on. Just POKE the code you need here to change the string printed.

While most printers use ASCII character codes and will respond properly, there are always some individual control and character codes. The program will probably need customizing to allow their use. The codes output in response to Sinclair graphic characters (Sinclair codes 1-10) are held in sequence beginning at address 16663. Those resident are all control codes. Inverse number codes are in sequence from address 16690. Those are set up to print characters not on the ZX-81 keyboard. (See Table 3 for conversions.) To add codes for your printer, just POKE the decimal value you want output to the address required (avoid POKES outside those two areas).

One last possible modification: The program automatically outputs a "linefeed" after a "carriage return." If your printer is set up to require just the "carriage return," it will print double-spaced lines. To correct that in the program, POKE 16623,24.

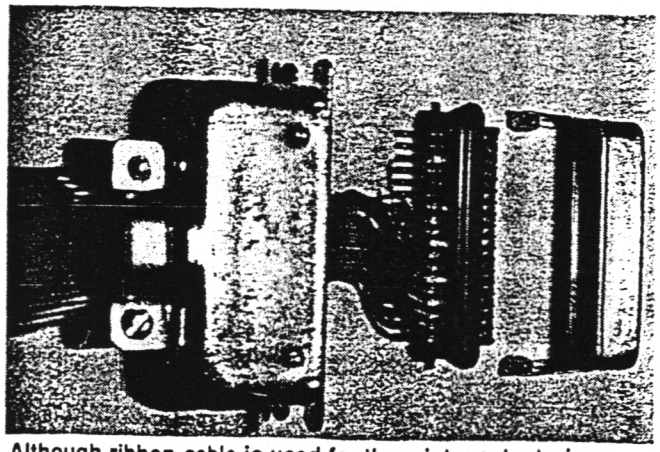
As you can see, that is something different from the LPRINT command, and although not as simple as LPRINT to use, the program is far more versatile.

Existing programs should be easily modified once the code is resident in the first line. Just replace the LPRINT statements with a string definition (don't forget to use control codes, particularly a carriage return): the RAND USR 16529.

The first time you see the printer busy taking orders form this "toy" computer could be slightly shocking. You won't believe that something so small can tell a big printer what to do. It gives you a new respect for the power of those little marvels. ■

TABLE 3—CHARACTER CONVERSION

Sinclair CHR$ Code	Output to Printer Hex	Output to Printer Decimal	Output Address
Graphics			
1	09	9	16663
2	0E	14	16664
3	14	20	16665
4	0F	15	16666
5	12	18	16667
6	1B	27	16668
7	08	8	16669
8	0D	13	16670
9	0A	10	16671
10	0C	12	16672
Inverse #'s			
156	40	64	16689
157	27	39	16690
158	5F	95	16691
159	26	38	16692
160	21	33	16693
161	22	34	16694
162	7C	124	16695
163	25	37	16696
164	5B	91	16697
165	5D	93	16698



Although ribbon cable is used for the printer output wires, use a solder-type Centronics connector.